Test Magic Design Document

**Table of contents**

[1 Introduce 3](#_Toc407029818)

[2 Architecture 4](#_Toc407029819)

[2.1 Big picture 4](#_Toc407029820)

[2.2 Station 4](#_Toc407029821)

[2.3 Master 4](#_Toc407029822)

[2.4 Master – Station communication 4](#_Toc407029823)

[3 Detailed design 5](#_Toc407029824)

[3.1 Station 5](#_Toc407029825)

[3.2 Master 5](#_Toc407029826)

**Figures** **list**

[Figure 1 - Master - Station model 6](#_Toc407029985)

**Table list**

**No table of figures entries found.**

# Introduce

The Test Magic is an improvement of FATE (FPT Automation Test Embedded) project. Which is limitation in flexibility and end-user familiar.

FATE framework have following disadvantages:

* Test case was fixed on client when it created. When a PC that have planned got a problem and cannot join the plan anymore, we must setup another PC to replace. When a PC is idle, we cannot attach it into plan to share other PCs task.
* Do not flexible in test information management.
* Difficult in using & controlling.
* Database size is fast scaling.

Test Magic is more powerful automation test framework with the following main features:

* Use master – slaves model. One master on one test plan, the master responsible for manages registered slave (station), assign task for the slave depends on slave’s resource, and detects the slave connection problem.
* Flexible scheduler. Scheduler is managed by the master. The jobs in scheduler will be assigned to a corresponding slave, depends on slave’s resources. When a PC disconnected due to any network/hardware problem, the disconnected PC’s jobs will be assigned to another PC.
* Easy to control. With XMLRPC protocol, the master and slaves can be controlled from anywhere through web site or software.
* Easy test plan creator. User can easily create the test plan by update test plan information in XML/text file, then upload this file to the master, the master will create the test plan. User can view the test plan and edit directly by website if needed.
* Test plan information will be in-dependent. Each test plan and results are stored in one separated database, so when user want to re-run the test for any version in the pass, the test case information is same as the past.

# Architecture

## Big picture



Figure 1 - Network model

The master could run from any PC, and this PC will called master.



Figure 2 - Master - Station communication

## Station

## Master

## Master – Station communication

# Detailed design

## Station

## Master